

Patent Claims

- 20043
- 5
1. Method for a fast performance of network operations via a network having high delay times by means of a module for processing a system call of an application layer and for initiating network operations of a network layer, with the following steps:
- transmission of the system call to the module,
 - determination of an execution mode of the system call by differentiating between a blocking and a non-blocking execution mode,
 - direct return of a logical value to the application layer and initiation of a network operation in the case of a non-blocking performance mode.
- 10
- 15
2. Method according to claim 1, wherein the network operation is transmitted to a partner instance communicating with a unit initiating the network operation.
- 20
3. Method according to claim 2, wherein the network operation received in the partner instance is converted into an operation, which is performed, and wherein a result of the operation is returned to the unit, that initiated the network operation.
- 25
- 30
- 35
4. Method according to any of claims 1 to 3, wherein a processing of the received result of an operation is realized in the module.
5. Method according to any of claims 1 to 4, wherein, upon the initiation of the network operation a non-blocking system call is converted into a state, in which an actual result of the system call executed in a partner instance is awaited without blocking the execution of the calling application.

November 15, 2000

6. Method according to any of claims 3, 4 or 5, wherein the received results refer to a non-blocking state and have a logical value, or are a result of a blocking system call executed in the partner instance.
7. Method according to any of claims 3 to 6, wherein the received results with a non-blocking execution mode are buffered.
8. Method according to any of claims 1 to 7, wherein the logical values either have a logical positive or a logical negative propositional value.
9. Method according claim 8, wherein the logical negative results are reported to the application with the execution of the following system call in the form of a logical negative return value.
10. Method according to any of claims 3 to 9, wherein, with a non-blocking system call, in the case of non-pending negative results of previous calls always a logical positive value is returned to the application.
11. Method according to any of claims 1 to 10, wherein the last system call of a connection is set into a blocking state in order to guarantee a return report of the results of the previously performed operations.
12. Method according to claim 1, wherein the blocking system calls are realized in a known manner by waiting for the result of the system call executed in the partner instance.
13. Method according to one of claims 1 to 12, wherein the system calls are socket system calls.

November 15, 2000

Sub 94 14. Method according to claim 13, wherein the socket system calls form a programming interface for an operating system.

5 15. Method according to claim 1, wherein the module is a pipeline module.

10 16. Device for a fast performance of network operations via a network having high delay times by means of a module for processing system calls of an application layer and for initiating network operations of a network layer, with

- a determining element for determining the execution mode of a system call,
- converting means for converting the system call into network operations,
- a sender for sending network operations,
- a receiver for receiving results of the performed network operations,
- a memory for storing received results,
- a processing element for processing the received results.

15 17. Device according to claim 16 with elements for realizing a state, in which the result of the system call executed in a partner instance is awaited once the network operation is initiated, without blocking the execution of the calling application.

20 18. Device according to claim 16 or 17, wherein the means for processing the received results differentiates between negative and positive values.